(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
21 November 2002 (21.11.2002)

PCT

(10) International Publication Number
WO 02/093414 A1

(51) International Patent Classification[7]: G06F 17/21, 17/30

(21) International Application Number: PCT/SG01/00089

(22) International Filing Date: 11 May 2001 (11.05.2001)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant *(for all designated States except US)*: **KENT RIDGE DIGITAL LABS** [SG/SG]; 21 Heng Mui Keng Terrace, Singapore 119613 (SG).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **WONG, Lung, Hsiang** [MY/SG]; Block 902, Jurong West Street 91, #04-109, Singapore 640902 (SG). **LOOI, Chee, Kit** [SG/SG]; Block 112, Potong Pasir Avenue 1,, #05-670, Singapore 350112 (SG).

(74) Agent: **GREENE-KELLY, James, Patrick**; Lloyd Wise, Tanjong Pagar, P.O. Box 636, Singapore 910816 (SG).

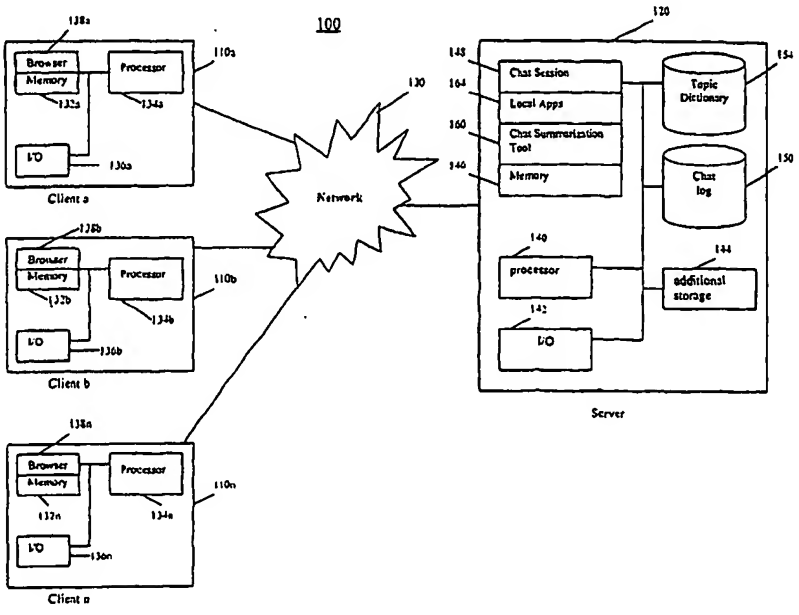
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: SYSTEM AND METHOD FOR CLUSTERING AND VISUALIZATION OF ONLINE CHAT



(57) Abstract: This invention provides a system and method of creating temporal clusters of content included in a chat session and representing such clusters in a chat timeline. The chat timeline provides a visul summary of the content of the chat session. The system further incldues additional user-friendly features that allow a user to view statistics of particular clusters, patterns of the chat session, and other relevant chat information.

WO 02/093414 A1

SYSTEM AND METHOD FOR CLUSTERING AND VISUALIZATION OF ONLINE CHAT

FIELD OF THE INVENTION

This invention relates generally to data processing systems and, more specifically, to systems and methods for clustering and visualization of content included in online chat sessions.

BACKGROUND OF THE INVENTION

The popularity of online chatting has increased significantly over the last several years. Online chat sessions may either be very structured and targeted to discussion of a particular topic, or may be unstructured and serve as an open forum for a public discussion of various topics. Unstructured online chat sessions are generally noisy and out-of-focus. The term "noise," as used herein, refers to off-topic content included in a topic-based online chat session. Participants in an online chat session, "chatters," may switch discussion topics frequently and randomly without alerting other users. The randomness and frequency with which topics may be changed makes it difficult for chatters to determine both a current topic and to trace previous content of the chat session. Some chat systems interface with a logging feature that allows users to log a chat transcript, indicating discussion topics covered in a particular chat session. The chat transcript is generally saved as an archive file and thus is not organized according to, for example, a threaded structure that can be easily indexed and searched. Such spontaneity and randomness in conventional online chat sessions thus poses several problems for chatters.

First, a latecomer to a particular chat session, or an existing chatter who leaves the chat session for a period of time and later returns, may have some difficulty tracing the previous discussion. Therefore it is hard for such person to join, or rejoin, the chat session. Second, individual chatters who enter a chat room when it is empty need a manner of determining a current status of a chat session. Conventional logging systems, as described above, are insufficient for this purpose. Similarly, even in chat sessions that include a logging feature that tracks chat discussions in the form of, for example, a chat transcript, searching such transcript to find a quality exchange can be a tedious and time consuming process and therefore discouraging to many users. Fourth, for topic-based

discussions, chatters are likely to initiate, perhaps inadvertently, off-topic discussions unless there is a mechanism that can detect the switching of topics and possibly provide a warning or alert about the topic switch, either directly or indirectly.

Accordingly, a need exists for an online chat analysis tool that tracks a chat session such that chatters may enter and exit the conversation randomly, and easily determine both the previous content of the chat session and the current status of the chat session. Additionally, for topic-based chat sessions a tool is needed to monitor a chat session to ensure that topic switching is avoided.

SUMMARY OF THE INVENTION

According to an embodiment of this invention, a method is provided to analyze chat content. The method includes determining a next utterance in a chat session, extracting keywords from the next utterance, associating the extracted keyword with a topic, and creating a cluster of the content of the chat session by organizing the content according to the topic.

According to another embodiment of this invention, a system is provided to analyze chat content. The system includes a chat summarization tool that divides a content included in a chat session into time-based segments of related chat data and provides a visual representation of the time-based segments of related chat data.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 depicts an illustrative block diagram of the invention.

Fig. 2 depicts an illustrative flow diagram of the invention.

Fig. 3 depicts an illustrative data structure that is used to facilitate logging of a chat session.

Figs. 4A-4E depict a chat timeline and illustrate various features of the chat timeline.

DETAILED DESCRIPTION OF THE INVENTION

This invention provides a chat session summarization tool that makes associations among and organizes the content of a chat session. The tool outputs a graphical representation of the content included in the chat session. This representation is provided in the form of a bar chart that depicts various utterances of the chat session and indicates when each of the utterances occurs relative to other utterances of the chat session, and a length of time each utterance occurred. In the visual representation of the chat session, the utterances are grouped as clusters, described further below. This chat summarization tool assists current chatters, newly joined chatters, and chat log readers to conveniently and easily determine a current status of a chat session and navigate a content history of the chat session. This chat summarization tool also creates a searchable chat database, i.e., a chat log, that allows users to search for segments of a chat session by topic or keyword.

More specifically, this invention analyzes the content of a chat session by associating utterances of a chat session with one another to create "clusters" of chat content. A "cluster" refers to a group of adjacent utterances with similar or related discussion topics in a chat session. Chat clustering includes applying a content analysis tool to the content of a chat session, identifying "clusters" of temporal chat utterances in the chat session, and grouping the clusters. Chat clusters may be separated, for example, by noise, i.e., off topic utterances of the chat session. Threshold parameters, configurable by users, determine the tolerance level of such "noisy utterances." Exemplary threshold parameters are described further below. An utterance may involve more than one topic, e.g., the discussion of "computers in education" involves two topics: "computer science" and "education". Therefore, multiple topics can be included in a single utterance. In addition to determining clusters with specific topics, the system can also detect "socializing clusters," including "looking for chatters," "greetings," and "separation." Socializing clusters are clusters that do not involve any specific discussion topic but are common socializing behaviors between humans, e.g., to exchange "hellos" or "goodbyes."

In the invention, the content of a chat session is divided into clusters according to a temporal sequence of the discussion topics. By contrast, conventional clustering methods are typically applied to a group of documents or items. The system of the

invention may execute either in real-time, i.e., while a chat session occurs, or after a chat session has completed.

Once the content of a chat session has been divided into clusters, a chat timeline is constructed. The chat timeline depicts a visual, temporal representation of chat topics included in the chat session. On the chat timeline, each cluster is represented as one or more lines of color. Each of the clusters that relate to a particular topic is represented by the same color. The length of each line corresponds to the time span of the content represented by the cluster. The cluster lines on the timeline are selectable, allowing users to conveniently view statistics of each cluster and read the chat log on a newly-launched window. Users may change the time span and the scale reflected on the timeline. Users may also change the temporal resolution of the timeline, e.g., on a scale of 1 minute or 10 minutes per unit on the timeline. The timeline construction may be performed by either a software program that interfaces with the chat room, or a software agent who has the access to the chat content. The chat timeline can either be updated in real-time during a chat session or after a chat session has completed if a chat session includes a logging feature.

This invention may also support additional embodiments, including, for example:

(1) tools for "pasting" notes, that will allow users, for example, to input comments on the content of a cluster or launch a new private chat session for further discussion on the content of a current chat session;

(2) updating chatters on topic changes and further including a feature to notify a chatter who is participating in a topic-based discussion when the chatter changes the topic;

(3) construction of a cluster search engine that allows a user to search a chat log by topic or keyword;

(4) detection of regular time-based patterns of discussion in a chat room. A time-based pattern of a chat room may include discussing a specific topic at a certain time on a particular day of the week. Thus, for example, an embodiment of this feature checks the contents of chat in a chat room during a specific day. This feature may be used to suggest to a user the best time to enter a particular chat room; and

(5) alerting a chatter when the chatter is off-topic, e.g., the system may display a pop-up window to inform a particular user that his or her utterances are off-topic relative to an ongoing discussion.

Fig. 1 depicts an illustrative block diagram of the invention. Clients 110 a ... n interface with server 120 via network 130. Each of clients 110 a...n includes the conventional components of memory 132, processor(s) 134, input/output devices 136, and browser 138. Server 120 includes the following conventional components: processor 140, input/output device 142, storage 144, and memory 146. Server 120 further includes additional storage, such as, chat log 150 and topic dictionary 154. Memory 146 may further include local application 148. Chat session summarization tool 160 analyzes the content of chat session 164 among clients 110 a...n and provides a visual representation of such content.

One of skill in the art will appreciate that while network 100 has been depicted with specific components, additional or different hardware or software components may be used within the scope of the invention. For example, topic dictionary 154 and chat log 150 may not reside in server 120, but may reside on a network accessible data storage device. Similarly, server 120 may not include local applications 148. Further, while chat summarization tool 160 has been depicted in memory 146, it may reside in a storage device connected to server 120 via a network. Still further, the processing described relative to chat summarization tool 160 may be performed by an attached procedure of the chat room or a software agent who has access to the chat content.

Fig. 2 depicts an illustrative flow diagram of the invention. Chat clustering is performed relative to 210-240 and 260; 250 relates to constructing a chat timeline. The invention first determines a next utterance in a chat session (210). An utterance refers to a single word or group of words or sentences entered by a specific chatter during a single entry. Thus, an utterance refers to a message that is created and sent to a chat room when the chatter presses the "enter" key.

Fig. 3 depicts an illustrative data structure that is used to facilitate logging of a chat session. The data structure of Fig. 3 depicts a linked list that defines a log unit and a cluster. An utterance is a log unit that is included as part of a cluster. As described above, an utterance may include multiple topics. For each utterance, the system extracts keywords by applying standard text parsing and morphology algorithms.

Once an utterance has been defined, keywords are extracted from the utterance (220). The key words correspond to words that have been included in a topic dictionary, such as topic dictonary 154. Each keyword that has been extracted from an utterance is then associated to one or more topics listed in the topic dictionary (230). Thus, each utterance is associated to a list of topics. For this purpose, the invention includes a "dictionary" that associates, i.e., maps, keywords to topics (235). The dictionary may be in the form of, for example, a relational database or a table, that lists keywords and related topics for each keyword. Each keyword can be associated to multiple topics. For example, the keyword "Java" can be associated to "programming languages," "geography," and "food & beverage." This dictionary may be in the form of, for example, a relational database or a look-up table that maps keywords to topics. Once a keyword has been mapped to one or more topic, the topics are categorized and organized hierarchically.

The topics are categorized according to the following logic. If a keyword can only be mapped to one topic, then the keyword is added to the topic list associated to the utterance that the keyword was extracted from; If the keyword can be mapped to more than one topic, then the system determines whether the topic has been included in a recent chat cluster. Recent is determined according to user-specified values reflecting constraints of a cluster, which are described further below. If there are one or more topics that relate to a recent chat cluster(s), then each of the topic(s) is added to a topic list associated with the utterance; If the topic does not relate to a recent cluster, then the system determines whether the particular keyword, or other keywords (if any), in the utterance relate to the same topic; If the topics share a particular keyword, then the systems adds the topic to the topic list of the utterance; If no such topic is found, then the systems adds the topic with the highest corresponding confidence rating (described further below) to the topic list of the utterance.

The topics are organized hierarchically according to a "confidence rating" between zero and one that is assigned to each keyword-topic mapping. If a keyword is associated to more than one topic, each mapping may possess different "confidence ratings," depending on which mapping is more likely to happen. For example, a chat room with a computer theme may assign the highest rating to the mapping of "Java → programming_languages"; whereas a chat room with a Southeast Asian-theme may give "Java → geography" the highest rating. A dictionary editor is provided for the chat room

administrator to add/delete topics, re-arrange the topic hierarchy, add/delete keywords, and edit confidence rates.

Once the topic list for an utterance is generated, the system divides the utterance into clusters (240). The system executes chat clustering recursively. In each recursion, the system focuses on a single utterance. More specifically, for real-time clustering, it handles the latest utterance; and for post-processing, it handles each utterance in chronological order. After the topic detection process is performed relative to a specific utterance, the system determines whether the specific utterance relates to a recent cluster(s). If the utterance does not relate to one or more recent clusters, a new cluster that includes the utterance is created. Hence, the system compares the topics with recent chat clusters to determine the most likely topic that the keyword in the utterance refers to.

In this processing, the system groups a current utterance with a recent cluster that relates to the same topic. When clustering an utterance, the system considers the following three user-specified parameters that indicate how "fine-grained" the clustering should be, i.e., how many utterances should be included in a cluster and the relationship among utterances included in a cluster. The size of an individual cluster is therefore influenced by specific threshold parameters. Bigger threshold parameters will result in the generation of clusters of larger sizes, which may involve more noise (i.e., off-topic utterances in individual clusters. The system ensures that each cluster satisfies the constraints posed by each of these threshold parameters. The threshold parameters include:

(1) Utterance Count Threshold (UCT): The *minimum* count of utterances needed to form a cluster;

(2) Utterance Proximity Threshold (UPT): The *maximum* count of off-topic utterances between a current utterance and a last utterance of a cluster that the cluster is allowed to be expanded to the current utterance; and

(3) Time Threshold (TT): The *maximum* time gap between a current utterance and a last utterance of a cluster that the cluster is allowed to be expanded to the current utterance.

After the utterance has been divided into clusters, the system generates and/or updates, as appropriate, the chat timeline (250). The chat timeline reflects a temporal

depiction of content of a chat session. Each "track" in a chat timeline consists of one or more "cluster lines" of the same topic, illustrating when and how long each topic is covered in the course of a chat session. Overlapping clusters/topics can thus be easily observed from the chat timeline. A user can select a cluster depicted on a chat timeline to gain additional information about the cluster and the utterances included in the cluster, described further relative to Fig 4, below.

After the chat timeline is updated, the system returns to 210 to analyze the next utterance and the processing repeats.

Example:

The following example depicts a process for clustering chat content and steps through exemplary algorithms that may be used to perform such processing. More specifically, the example demonstrates how the threshold parameters are used to detect clusters.

Utterance Index	Timestamp (hr:min:sec)	Topic(s) Detected
1	1:00:01	A
2	1:00:02	A
3	1:00:04	A, B
4	1:00:04	A, B
5	1:00:04	B
6	1:00:09	B
7	1:00:11	
8	1:00:11	
9	1:00:12	A
10	1:00:13	A
11	1:00:17	A, B, C
12	1:00:18	B
13	1:00:19	
14	1:00:21	
15	1:00:29	C
16	1:00:29	C
17	1:00:31	C

Case 1: UCT = 3; UPT = 5; TT = 10 seconds
Clusters formed:

Topic	Range (Utterance Indices)	Remarks
A	1-11	The cluster has a gap between 5-8. The gap is taken as part of the cluster because it only covers 4 utterances (less than UPT) and spans 8 seconds (less than TT).
B	3-12	The cluster has a gap between 7-10. The gap is taken as part of the cluster because it only covers 4 utterances (< UPT) and spans 8 seconds (< TT).
C	15-17	Utterance 11 is not taken as part of the cluster because the gap between 11 and 15 will cover 3 utterances (< UPT) but spans 11 seconds (> TT).

The first two clusters are overlapping with each other.

Case 2: UCT = 3; UPT = 3; TT = 12 seconds
Clusters formed:

Topic	Range (Utterance Indices)	Remarks
A	1-4	
A	9-11	The reason this cluster can't merge with the first cluster (1-4, Topic A) is that it will otherwise form a gap (5-8) that spans 8 seconds (< TT) but covers 4 utterances (> UPT).
B	3-6	The reason utterances 11 & 12 can't be included in this cluster is that it will otherwise form a gap (7-10) that spans 8 seconds (< TT) but covers 4 utterances (> UPT). Neither can utterances 11 & 12 make up a separate cluster because it only has 2 utterances with the same topic (< UCT).
C	11-17	The cluster has a gap between 12-14. The gap is taken as part of the cluster because it only covers 3 utterances (< UPT) and spans 11 seconds (< TT).

The 2nd and the 4th clusters are overlapping with each other.

The following is an illustrative algorithm to cluster chat content:

```
ALGORITHM clustering
INPUT: chat_log, UCT, UPT, TT;

VARIABLES:
log_unit current_utterance, current_utterance2;
cluster          eligible_block;
list of clusters eligible_clusters;
integer          count;

START
WHILE there are more unanalyzed utterances
        current_utterance := the first unanalyzed utterance;
        Analyze current_utterance ⇒ Form topic_list;
        Identify eligible_block (a temporary cluster of the preceding utterances who fit the following
            conditions: utterance count ≤ UPT AND time spanned ≤ TT);
        eligible_clusters are clusters that overlap with eligible_block;

        /* merge current utterance with an existing cluster */
        WHILE there are more unchecked clusters in eligible_clusters
                current_cluster := the first unchecked cluster in eligible_clusters;
```

```
                IF ∃ (topic in current_utterance's topic_list) that tallies with current_cluster's topic
                        Expand current_cluster to current_utterance (i.e., current_utterance is now part
                          of current_cluster);
                ENDIF
        ENDWHILE

        /* generate new cluster */
        WHILE there are more "unclustered" topics in current_utterance
                count := 1;
                current_topic := first unchecked topic of current_utterance;
                Identify eligible_block with respect to current_utterance (a temporary cluster of the
                    preceding utterances who fit the following conditions: utterance count ≤ UPT AND
                    time spanned ≤ TT);
                WHILE (there are more unchecked previous utterances) AND (there are more unchecked
                    utterances in eligible_block)
                        current_utterance2 := the closest unchecked utterance to current_utterance;
                        IF topic_list in current_utterance2 contains current_topic
                                count = count + 1;
                                IF (count = UCT)
                                        Generate new cluster with topic := current_topic and covers
                                            current_utterance2 (starting) to current_utterance
                                            (ending);
                                        EXIT (innermost) WHILE loop;
                                ELSE
                                        Identify eligible_block with respect to current_utterance2;
                                        NEXT (innermost) WHILE iteration;
                                ENDIF
                        ENDIF
                ENDWHILE
        ENDWHILE

        Refine the clustering;
        Update timeline display;
ENDWHILE
END
```

After a cluster is created, it is refined. The term "refining" a cluster refers to a cluster B being absorbed by, i.e., combined with, a cluster A when each of the following conditions is met:

(1) clusters A and B both overlap each other; AND

(2) cluster B's topic is a sub-category of cluster A's topic; AND

(3) the length, i.e., utterance count, of cluster B is less than 1/5 the length of cluster A.

A list of clusters, i.e., a cluster list, included in a particular chat session is stored in a chat log, such as, for example, a database. When a user, for example, the owner or the moderator of the chat room, changes the values of UCT, UPT and TT, the system

deletes a current cluster list and re-creates a new cluster list. Before modifying the UCT, UPT, and TT values, the user can save current cluster list.

Figs. 4A-4F depict a chat timeline and illustrate various features of the chat timeline. Each topic of clusters is depicted with a different color. A user may navigate the timeline, for example, via buttons provided on a user interface that allow a user to scroll left and right. Similarly, the user interface may include a scrollbar that allows a user to zoom in and out. Fig. 4B depicts a chat session that includes more tracks than can be displayed, for example, on a single page. In such case, the visualization tool combines "tracks" that do not have overlapping clusters. Each topic is still represented by a different color.

When a user clicks on a cluster line, a pull-down menu appears, see Fig. 4C, enabling the user to access to a series of features related to the cluster or the corresponding topic. For example, the following additional system features can be accessed via selection of a cluster:

(1) Display of a topic information window that displays the "accumulative" statistics and information of each cluster that has been associated to a particular topic, e.g., total time elapsed, active chatters, regular patterns, etc., and the starting and ending time of each cluster.

(2) Display of a cluster information window that displays statistics and other information related to the cluster. The other information may include, for example, time elapsed, a list of all participants in a chat session, i.e., chatters, active chatters, an indication of a quality of the conversation, i.e., a measure of how much noise, or off-topic discussion is included in a chat session (see Fig. 4D).

(3) Display of a chat log window displays the chat log of the cluster (see Fig. 4D).

(4) Access to an annotation tool that allows a user to add annotations/comments to a particular cluster. The annotations/comments are stored relative to individual clusters and can be displayed when a user requests to retrieve them. The annotation window is presented as a single-threaded online message board. If the user changes the values of UCT, UPT and TT, i.e., initiates a re-clustering of a chat session, the system will automatically attach the annotations to the new cluster(s) at the same portion of the dialogue with the same topic. The chat log is also provided with new cluster assignments when re-clustering occurs. For example, compare "Case 1" and "Case 2" in the above example.

(5) Launch of and access to a private chat room that is accessible to the chatters in the main chat room to engage in a topic-specific discussion directed to the topic reflected by the selected cluster. Each cluster can have one related private chat room.

(6) Entry into a private chat room to determine a status of the private chat room, by viewing, for example, a chat log or chat timeline, etc., and join a current discussion in the private chat room.

Figure 4E depicts a chat timeline that includes small icons corresponding to cluster annotations and private chat rooms. The icons are superimposed on the clusters having user annotations and/or private chat rooms. Thus, by double-clicking on an icon, a user may access annotations of a particular cluster, e.g., notes or statistics related to the cluster, or a private chat room. Figure 4F shows an example of an annotation on a cluster.

As described above, the system supports a search engine embodiment that allows a user to search a chat log for either specific topics or specific keywords. In this embodiment, when the user specifies a topic, the system lists the clusters which relate to, i.e., have been associated to, the topic. The system may also list the sub-categories that relate to the topic, which subcategories are configurable by the user, as links so to the relevant portions of the chat log and its statistics. If the resultant list is too large, the user can narrow it down by specifying more conditions, such as, for example, keywords of a cluster, time periods of a cluster, or participants in a chat session or specific cluster of a chat session, etc.

Although this invention has been described relative to a particular embodiment, one of skill in the art will appreciate that this description is merely exemplary and the system and method of this invention may include additional or different components. This description is therefore limited only by the appended claims and the full scope of their equivalents.

WHAT IS CLAIMED IS:

1. A method to analyze chat content, comprising:

determining a next utterance in a chat session;

extracting keywords from the next utterance;

associating the extracted keywords with a topic; and

creating a cluster of the content of the chat session by organizing the content according to the topic.

2. The method of claim 1, further comprising creating a chat timeline reflecting the content of the chat session by cluster.

3. The method of claim 2, wherein creating the timeline comprises creating a temporal representation of the content of the chat session.

4. A system to analyze chat content, comprising a server that includes a chat summarization tool that divides content of the chat session into time-based segments of related chat content and provides a visual representation of the time-based segments of the related chat content.

5. The system of claim 4, further including a client that participates in an online chat session.

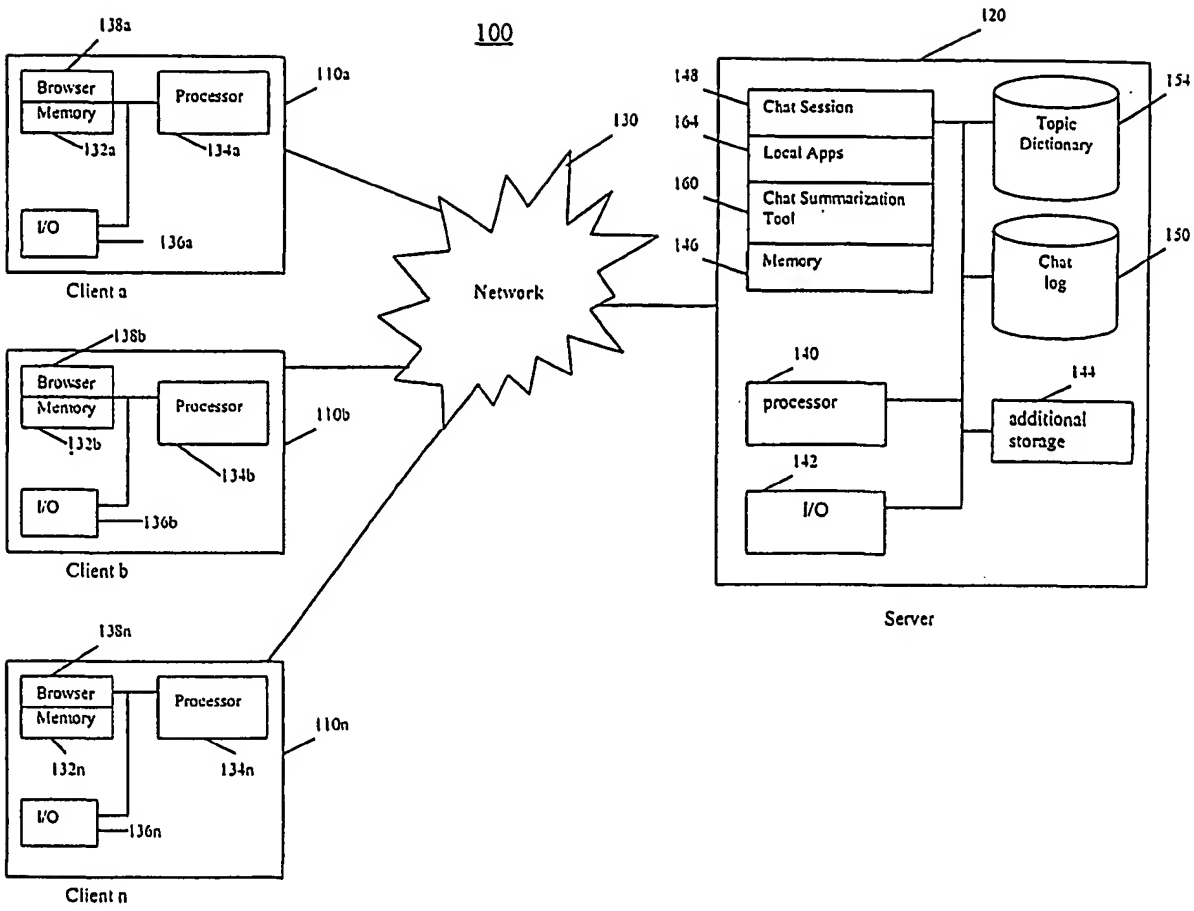

100
138a
Browser
Memory
Processor
110a
132a
134a
I/O
136a
Client a
138b
Browser
Memory
Processor
110b
132b
134b
I/O
136b
Client b
138n
Browser
Memory
Processor
110n
132n
134n
I/O
136n
Client n
Network
130
120
148
Chat Session
164
Local Apps
160
Chat Summarization Tool
146
Memory
154
Topic Dictionary
150
Chat log
140
processor
144
additional storage
142
I/O
Server

Fig. 1

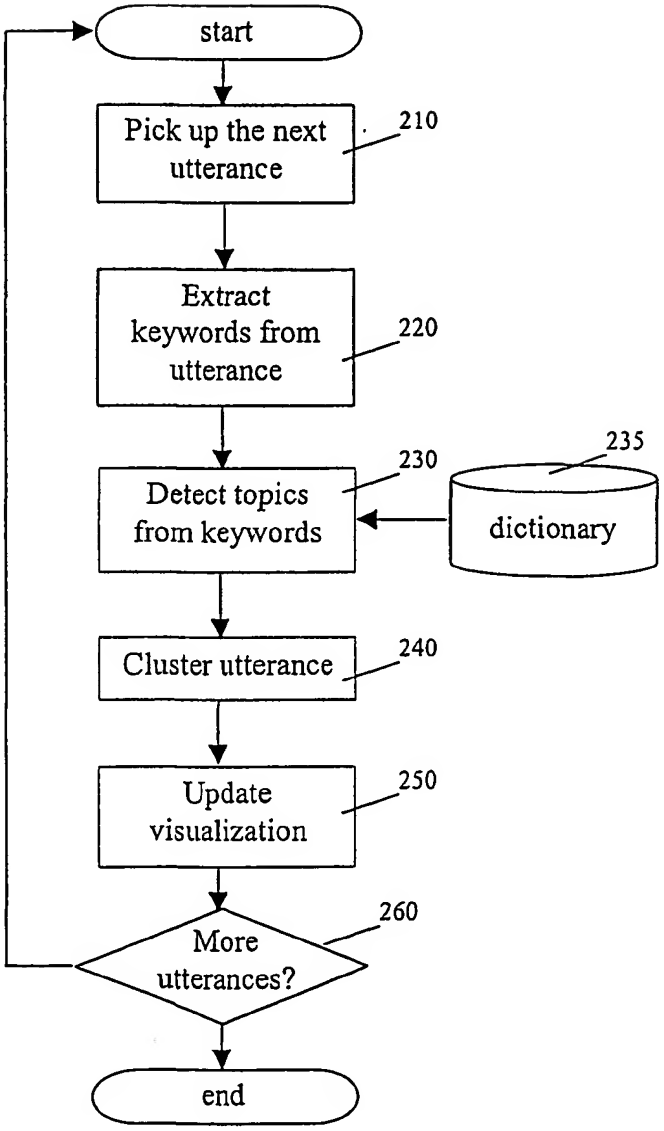

start
Pick up the next utterance
210
Extract keywords from utterance
220
Detect topics from keywords
230
235
dictionary
Cluster utterance
240
Update visualization
250
More utterances?
260
end

Fig. 2

linked_list: chat_log – list of **log_unit**

log_unit:

- integer: index_no
- date_time: time_stamp
- string: user_id
- action_type: action_type (ENTER, LEAVE, SAY, EMOTE)
- string: content (for SAY – utterance; for EMOTE – description of the emotion)
- **linked_list**: topic_list – list of strings

linked_list: cluster_list – list of **cluster**

cluster:

- string: topic
- integer: start_index_no
- integer: end_index_no

Fig. 3

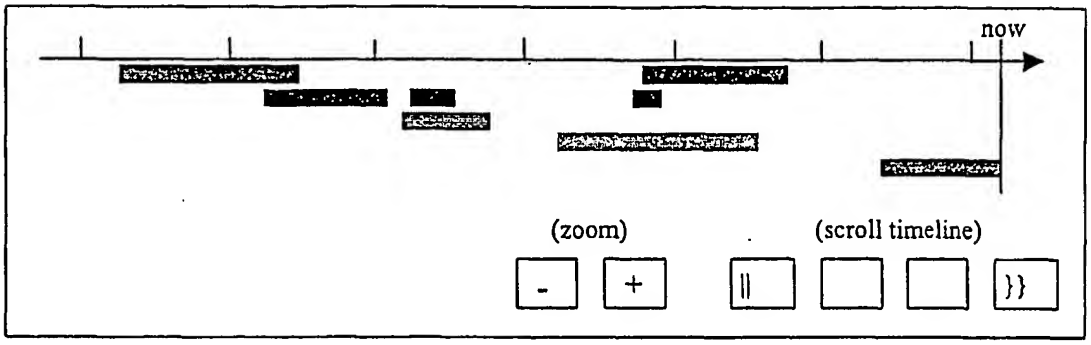


Fig. 4A

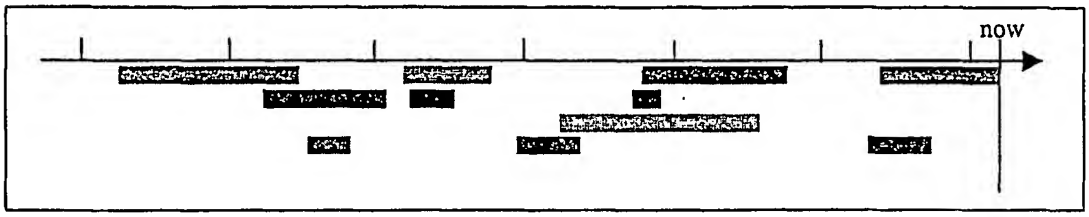


Fig. 4B

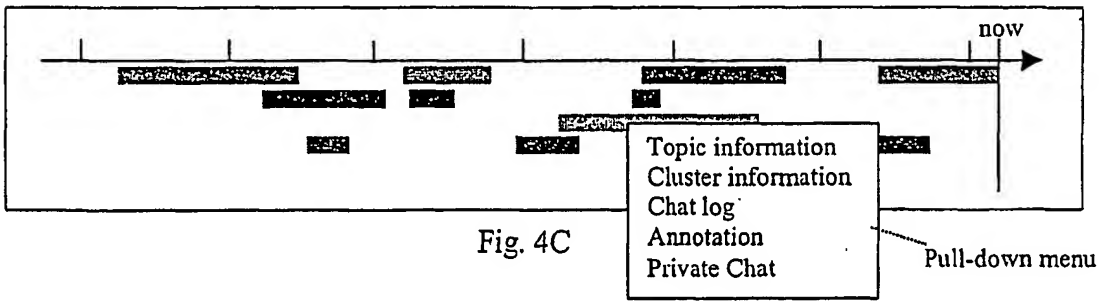


Fig. 4C

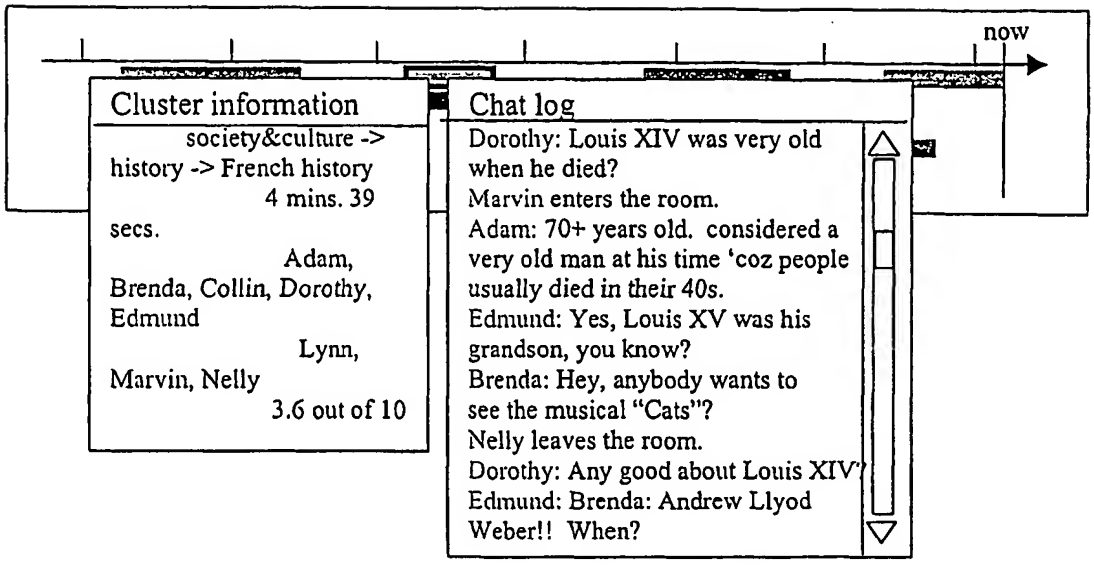


Fig. 4D

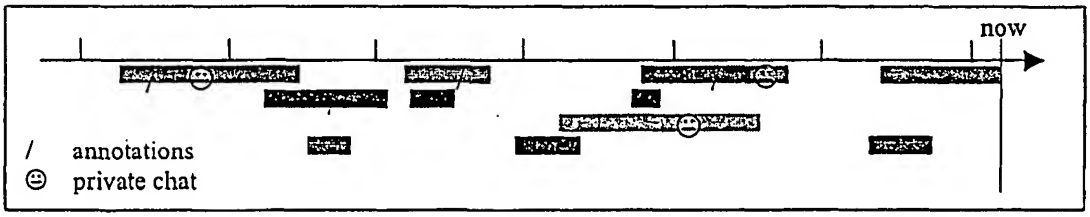


Fig. 4E

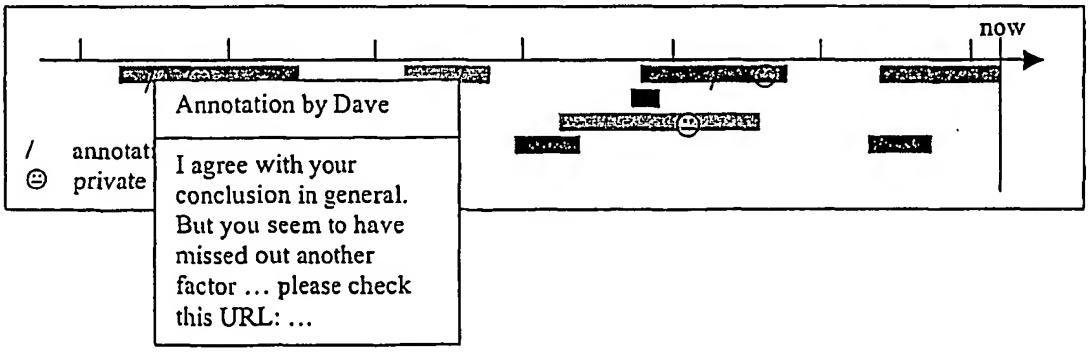


Fig. 4F